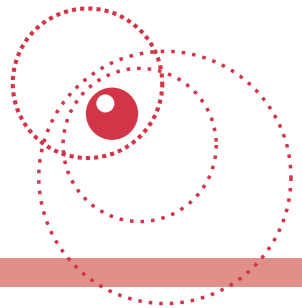


CS3236: Introduction to Information Theory

Lecture 13: Properties of Kolmogorov Complexity

April 22, 2026

Pierre Senellart

TELECOM ParisTech

Conditional Kolmogorov Complexity

- Let $x, y \in \{0,1\}^*$ be two binary sequences
- xy is not self-delimiting: impossible to determine where x ends and y starts
- We define $\langle x, y \rangle$ as $0^{\ell(x)}1xy$ to allow self-delimitation
- $\ell(\langle x, y \rangle) = 2\ell(x) + \ell(y) + 1$

Definition

Given a Turing machine T, the Kolmogorov complexity of y given z relative to T, where $y, z \in \{0,1\}^*$, is defined as:

$$C_T(y \mid z) = \min_{\substack{x \in \{0,1\}^* \\ T(\langle z, x \rangle) = y}} \ell(x)$$

Theorem (Invariance)

There exists a Turing machine U such that for any Turing machine T, there exists a constant $\gamma_{T,U}$ such that for any $y, z \in \{0,1\}^*$:

$$C_U(y \mid z) \leq C_T(y \mid z) + \gamma_{T,U}$$

Theorem (Invariance)

There exists a Turing machine U such that for any Turing machine T, there exists a constant $\gamma_{T,U}$ such that for any $y, z \in \{0,1\}^*$:

$$C_U(y \mid z) \leq C_T(y \mid z) + \gamma_{T,U}$$

We can then fix such a Turing machine U and write $C(y \mid z)$ with no subscript, results will hold for any such Turing machine, up to an additive constant.

Proof.

Let U' be a universal Turing machine with encoding function $\alpha_{U'}$ (e.g., $E(\cdot)$ as defined in the previous lecture), and U a Turing machine that, given input $\langle z, \alpha_U(T)x\rangle$ for Turing machine T and $x, z \in \{0,1\}^*$, executes U' on input $\alpha_{U'}(T)\langle z, x\rangle$.

Then for any Turing machine T and $y, z \in \{0,1\}^*$, let $x \in \{0,1\}$ be such that $\ell(x) = C_T(y \mid z)$ and $T(\langle z, x\rangle) = y$.

$$U'(\langle z, \alpha_U(T)x\rangle) = U'(\alpha_{U'}(T)\langle z, x\rangle) = T(\langle z, x\rangle) = y$$

and $\ell(\alpha_U(T)x) = C_T(y \mid z) + \ell(\alpha_U(T))$. □

Proposition

There exist constants γ, γ' such that for any $y, z \in \{0,1\}^*$:

$$C(y \mid z) \leq C(y) + \gamma \leq l(y) + \gamma'$$

Note that for any Turing machine T, for all $y, z \in \{0,1\}^*$, $C_T(y) \leq C_T(y \mid z) + 2\ell(z) + 1$. In particular $C(y) \leq C(y \mid \varepsilon) + 1$.

Definition

Let α be some fixed positive constant. A sequence $y \in \{0,1\}^*$ is α-incompressible if $C(x) \geq \ell(x) - \alpha$.

Definition

Let α be some fixed positive constant. A sequence $y \in \{0,1\}^*$ is α-incompressible if $C(x) \geq \ell(x) - \alpha$.

How many sequences are incompressible?

Theorem (Incompressibility)

Let m be a positive integer and $z \in \{0,1\}^*$. Let $0 \leq \alpha \leq \log m$ be an arbitrary real, such that $\log m - \alpha$ is an integer. Then every set $A \subseteq \{0,1\}^*$ of cardinality m has at least $m(1 - 2^{-\alpha}) + 1$ elements y such that $C(y \mid z) \geq \log m - \alpha$.

Theorem (Incompressibility)

Let m be a positive integer and $z \in \{0,1\}^*$. Let $0 \leq \alpha \leq \log m$ be an arbitrary real, such that $\log m - \alpha$ is an integer. Then every set $A \subseteq \{0,1\}^*$ of cardinality m has at least $m(1 - 2^{-\alpha}) + 1$ elements y such that $C(y \mid z) \geq \log m - \alpha$.

Extreme cases:

$$\begin{cases} \text{when } \alpha = 0: & \text{at least 1 element } y \text{ with } C(y \mid z) \geq \log m \\ \text{when } \alpha = \log m: & \text{at least } m \text{ elements } y \text{ with } C(y \mid z) \geq 0 \end{cases}$$

Proposition

For $y \in \{0,1\}^*$, let y^* be such that $\ell(y^*) = C(y)$ and $U(y^*) = y$. There exists a constant $\alpha > 0$ such that for all $y \in \{0,1\}^*$, y^* is α-incompressible.

Proposition

For $y \in \{0,1\}^*$, let y^* be such that $\ell(y^*) = C(y)$ and $U(y^*) = y$. There exists a constant $\alpha > 0$ such that for all $y \in \{0,1\}^*$, y^* is α-incompressible.

Proposition

There exists a constant $\alpha > 0$ such that for any positive integer n, there exist $y, z \in \{0,1\}^*$ with $\ell(y) \leq n$ and $\ell(z) \leq n$ such that:

$$C(\langle x, y \rangle) \geq C(x) + C(y) + \log n - \alpha$$

Theorem

Let $y = y_1 y_2 \ldots y_m \in \{0,1\}^{nm}$ be a binary sequence with, for $1 \leq k \leq m$, $y_k \in \{0,1\}^n$. For any $z \in \{0,1\}^n$, we write $\Pr_Z(z) := |\{1 \leq k \leq m : y_k = z\}| / m$, and Z the corresponding random variable.

Then, up to an additive constant independent of y:

$$C(y) \leq m(H(Z) + 2^{n+1}\ell(\bar{m})/m).$$

Theorem

Let $y = y_1 y_2 \ldots y_m \in \{0,1\}^{nm}$ be a binary sequence with, for $1 \leq k \leq m$, $y_k \in \{0,1\}^n$. For any $z \in \{0,1\}^n$, we write $\Pr_Z(z) := |\{1 \leq k \leq m : y_k = z\}| / m$, and Z the corresponding random variable.

Then, up to an additive constant independent of y:

$$C(y) \leq m(H(Z) + 2^{n+1}\ell(\bar{m})/m).$$

Proof.

Consider a Turing machine that takes as input the list of frequencies $s_k = p_k m$ as well as an index of the sequence y among all sequences with the same list of frequencies. □

Theorem

Let $Y^{(n)}$ be the random variable corresponding to n draws from an i.i.d. source Y. Then $\mathbb{E}(C(Y^{(n)})) \sim_{n\to\infty} H(Y^{(n)})$.

Shown for a uniform distribution on the binary source, see Section 8.1 of the textbook for generalization.

How to test for randomness of a coin toss sequence?

- Statistical tests
  - Test that about half the coin tosses are heads
  - Test that the sequence heads-heads occur in about 1/4th of the cases
  - Test that there are roughly as much heads in even places of the sequences than in odd places
  - ...

How to test for randomness of a coin toss sequence?

- Statistical tests
  - Test that about half the coin tosses are heads
  - Test that the sequence heads-heads occur in about 1/4th of the cases
  - Test that there are roughly as much heads in even places of the sequences than in odd places
  - ...
- One can show (Theorem 2.4.2 of the textbook) that any such (computable) test is captured by the following test:

$$\Pr(x \mid \ell(x) - C(x \mid \ell(x)) - 1 \geq m) \leq^? 2^{-m}$$

(Supervised) machine learning: given a set of examples, find a model that maximizes the likelihood of producing this set of examples

- (Supervised) machine learning: given a set of examples, find a model that maximizes the likelihood of producing this set of examples
- Problem: the best fitting model is the one that always produces this exact same of examples; no generalization!

(Supervised) machine learning: given a set of examples, find a model that maximizes the likelihood of producing this set of examples

- Problem: the best fitting model is the one that always produces this exact same of examples; no generalization!
- Usual fix: force the model to be of a certain form (e.g., an SVM, a linear regression, a neural network of a certain shape, etc.)

(Supervised) machine learning: given a set of examples, find a model that maximizes the likelihood of producing this set of examples

- Problem: the best fitting model is the one that always produces this exact same of examples; no generalization!
- Usual fix: force the model to be of a certain form (e.g., an SVM, a linear regression, a neural network of a certain shape, etc.)
- Problem: this choice is arbitrary. How to choose between two models of different forms?

(Supervised) machine learning: given a set of examples, find a model that maximizes the likelihood of producing this set of examples

- Problem: the best fitting model is the one that always produces this exact same of examples; no generalization!
- Usual fix: force the model to be of a certain form (e.g., an SVM, a linear regression, a neural network of a certain shape, etc.)
- Problem: this choice is arbitrary. How to choose between two models of different forms?
- Solution: Following Kolmogorov complexity, use the model such that the total description length of model + examples is minimal: minimum description length principle (or MDL)

- The program that has lowest length for an object gives the intrinsic description of an object
- Can model the "best" way to describe an object, the way a human brain would represent this object
- Examples:
  - Occam's razor: the most concise explanation is the most likely one
  - from a corpus of natural language sentences, build a minimum-description model of the natural language
  - comparative linguistics: compare $C(x \mid y)$ and $C(x \mid z)$ for two translations y and z in two different languages of a given text to determine which language is historically closer to the original one

Kolmogorov complexity is undecidable:

Theorem

There is no Turing machine T, such that, for any $y \in \{0,1\}^*$, $T(y) = \overline{C(y)}$.

Kolmogorov complexity is undecidable:

Theorem

There is no Turing machine T, such that, for any $y \in \{0,1\}^*$, $T(y) = \overline{C(y)}$.

... but can be approximated:

Proposition

For a Turing machine T and binary sequence $y \in \{0,1\}^*$, denote $T^{(n)}(y) := T(T^{(n-1)}(y))$ and $T^{(1)}(y) := T(y)$.
There exists a Turing machine T such that for any $y \in \{0,1\}^*$, $\lim_{n\to\infty} T^{(n)}(y) = \overline{C(y)}$.

- "Real" Kolmogorov complexity is out of reach
- But possible to define Kolmogorov complexity for simpler, non universal, computation models, for which minimum length description will be computable (e.g., regular expressions)
- Or use fancy compression algorithms as a proxy for Kolmogorov complexity
- Or more generally be ok with only having an approximation of Kolmogorov complexity

- I'm actually not an information theorist

- I'm actually not an information theorist

  My fields of research are: database theory (logical models for data representations and database queries) and Web mining (extracting valuable information from the Web)

- I'm actually not an information theorist
- My fields of research are: database theory (logical models for data representations and database queries) and Web mining (extracting valuable information from the Web)
- Still, many examples of uses of information theory in my research:

- I'm actually not an information theorist
- My fields of research are: database theory (logical models for data representations and database queries) and Web mining (extracting valuable information from the Web)
- Still, many examples of uses of information theory in my research:
  - Optimal schema mapping from one Web source to another: the one that minimizes description of the schema mapping and the corrections, for some restricted logical language

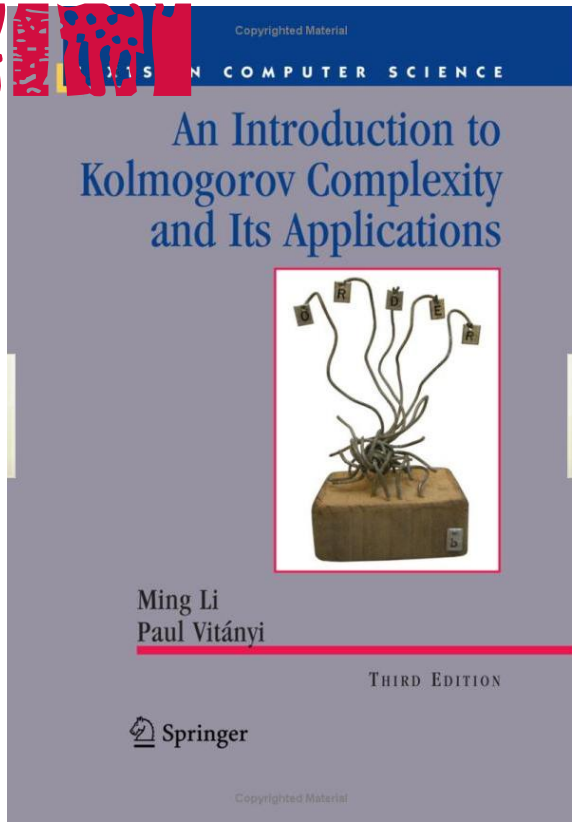
- I'm actually not an information theorist
- My fields of research are: database theory (logical models for data representations and database queries) and Web mining (extracting valuable information from the Web)
- Still, many examples of uses of information theory in my research:
  - Optimal schema mapping from one Web source to another: the one that minimizes description of the schema mapping and the corrections, for some restricted logical language
  - How to predict which tweets will get retweeted by whom? Those that have minimum Kolmogorov complexity given the user

- I'm actually not an information theorist
- My fields of research are: database theory (logical models for data representations and database queries) and Web mining (extracting valuable information from the Web)
- Still, many examples of uses of information theory in my research:
  - Optimal schema mapping from one Web source to another: the one that minimizes description of the schema mapping and the corrections, for some restricted logical language
  - How to predict which tweets will get retweeted by whom? Those that have minimum Kolmogorov complexity given the user
  - How to measure how informative the use of a term is on the Web, for ranking purposes? Use its information content

- I'm actually not an information theorist
- My fields of research are: database theory (logical models for data representations and database queries) and Web mining (extracting valuable information from the Web)
- Still, many examples of uses of information theory in my research:
  - Optimal schema mapping from one Web source to another: the one that minimizes description of the schema mapping and the corrections, for some restricted logical language
  - How to predict which tweets will get retweeted by whom? Those that have minimum Kolmogorov complexity given the user
  - How to measure how informative the use of a term is on the Web, for ranking purposes? Use its information content
  - What is the area of a Web page that is the most relevant to a set of keywords? The one which is simplest to describe in terms of these keywords

Possibility to follow a double Master's degree from NUS and a French elite school `http://www.fddp.nus.edu.sg/`

- Best school in France specializing in information technology
- Great research in:
  - All aspects of information theory (signal, electronic communications, cognitive sciences)
  - Statistical machine learning
  - Networking
  - And of course database theory and Web mining



Textbook on Kolmogorov complexity, sections 2.1–2.3, and 2.8 (not compulsory, but useful to go into more detail).

Previous homework and project due Friday night this week.